An Approach for Cross-Domain Intrusion Detection

Thuy Nguyen, Mark Gondree, Jean Khosalim, David Shifflett, Timothy Levin and Cynthia Irvine
Naval Postgraduate School, Monterey, California, USA
tdnguyen@nps.edu
mgondree@nps.edu
jkhosali@nps.edu
shifflett@nps.edu
levin@nps.edu
irvine@nps.edu

Abstract: Network-based monitoring and intrusion detection has grown into an essential component of enterprise security management. Monitoring potentially malicious activities across a set of networks classified at different security levels, however, presents subtle and complicated challenges. Analysis of intrusion alerts collected on an individual network only reveals malicious attempts to compromise that particular network, not the overall attack patterns across the enterprise. Development of a comprehensive perspective for intrusion analysis of all networks in a multilevel secure (MLS) environment requires care to ensure that the enforcement of information flow control policies is preserved. We describe an approach to cross-domain network-based intrusion detection. Leveraging the Monterey Security Architecture (MYSEA) high-assurance MLS federated computing framework, we developed an MLS policy-constrained network-based CD-IDS prototype using untrusted single-level components and multilevel (trusted) components, supported by open source software (i.e., BASE, snort, PostgreSQL and pgpool-II). Our prototype enables an analyst to view and manipulate network trace data collected from multiple networks, while enforcing mandatory access control policies to constrain the analyst to only those resources her session level dominates.

Keywords: cross-domain services, multilevel security, intrusion detection, quality of security service

1. Introduction

Technological strategies for mitigating network threats have become essential for any Enterprise. In particular, the DoD mandates both a firewall and network intrusion detection system (IDS) be components of any enclave boundary defense strategy (DoD 2003). The Defense Security Service highlights this requirement for Secret Internet Protocol Router Network (SIPRNET) enclaves, but the mandate applies equally to enclaves associated with any sensitivity level. Thus, the process of monitoring and analyzing network activity for anomalies or attack evidence is largely performed independently, for each network.

To our knowledge, however, the literature has not yet explored the problem of monitoring malicious activity across a variety of networks of different sensitivities. A number of trends, including the growing number of interconnected sensitive and tactical networks, create the realistic prospect of simultaneous, correlated, distributed attacks against multiple networks of different sensitivities. Such attacks may be coordinated via legitimate channels created by cross-domain guards (Bailey 2007)—*e.g.*, a unidirectional security gateway, or data diode, passing messages from low networks to high networks—or via some covert or out-of-band channel.

Regardless of the attack mechanism, analyzing and associating the attack evidence, given separate independent views of the "battlefield," poses a technical challenge to effective network threat detection. As an alternative, we posit that a *unified view* of all network intrusion alerts may better assist the analyst in deriving an attack pattern and formulating an effective defense strategy. Where networks operate at different sensitivity levels, data like IP addresses, traces, and other attack evidence from a network should be treated as being at the sensitivity of that network. It follows that an analyst should be restricted to a *multilevel view* associated with her clearance, following the notion of multilevel views formalized by Denning, *et al*. (Denning 1988). From this, we derive the basic premise of a *cross-domain intrusion detection system* (CD-IDS): an intrusion analysis engine providing an appropriate multilevel view of attack data from networks of different classifications, leveraging the single-level intrusion monitoring devices already present in those networks.

We note that, after an attack is detected, the problem of deploying countermeasures to an attack across networks of various sensitivities is similarly nontrivial. In particular, following the principle of non-interference, countermeasures deployed on low sensitivity networks must be agnostic to the

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2012		**00-00-2012 to 00-00-2012**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
An Approach for Cross-Domain Intrusion Detection	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Postgraduate School,Monterey,CA,93943**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Network-based monitoring and intrusion detection has grown into an essential component of enterprise security management. Monitoring potentially malicious activities across a set of networks classified at different security levels, however, presents subtle and complicated challenges. Analysis of intrusion alerts collected on an individual network only reveals malicious attempts to compromise that particular network, not the overall attack patterns across the enterprise. Development of a comprehensive perspective for intrusion analysis of all networks in a multilevel secure (MLS) environment requires care to ensure that the enforcement of information flow control policies is preserved. We describe an approach to cross-domain network-based intrusion detection. Leveraging the Monterey Security Architecture (MYSEA) high-assurance MLS federated computing framework, we developed an MLS policy-constrained network-based CD-IDS prototype using untrusted single-level components and multilevel (trusted) components, supported by open source software (i.e., BASE, snort, PostgreSQL and pgpool-II). Our prototype enables an analyst to view and manipulate network trace data collected from multiple networks, while enforcing mandatory access control policies to constrain the analyst to only those resources her session level dominates.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**10**	

presence of attacks on higher-sensitivity networks. On the other hand, it is very sensible for preventative defenses to be deployed on networks of higher sensitivities, based on the presence of attacks in neighboring networks of lower sensitivity.

The remaining sections describe the system architecture, the concept of operations, and the implementation of a CD-IDS prototype. The core components of the CD-IDS prototype are application-level services, running on a distributed MLS system. A user can invoke these services remotely (at the user's session level) to access data from any network whose classification is dominated by the user's session level. The paper concludes with a discussion of the challenges we encountered during development and of potential future work.

2. System Architecture overview

The Monterey Security Architecture (MYSEA) is a high-assurance, multilevel secure, distributed client-server system architecture which provides authenticated users secure access to data and services at different classification levels, using popular commercial off-the-shelf (COTS) applications running on stateless thin clients (Irvine 2009). We briefly describe the components and features of the current MYSEA system, as it serves as the platform for the CD-IDS prototype.

2.1 Architecture overview

MYSEA features a combination of (relatively few) specialized policy enforcing components and multiple open source and COTS components.

The federation of high assurance MLS servers provides the locus of security policy enforcement, while the highly trustworthy Trusted Path Extension (TPE) and Trusted Communications Module (TCM) components authenticate/disambiguate users and single level networks, respectively. Each is a gate-keeper for user interaction with the MYSEA Server. Other system elements afford users the ability to run unmodified office productivity tools, web-based services, and DoD applications. Federated servers support scalability, which, when combined with single sign-on and virtualization, result in an extensible computing environment. The major components of the architecture are shown in Figure 1 and include:

- High assurance MYSEA Servers, which together enforce the system-wide multilevel security policy and host various open source or commercial application services.
- Client workstations executing popular software applications; and TPEs, which intercept network traffic between the client and the MYSEA Server, providing trustworthy network security, identification and authentication, and policy support.
- Existing single level networks connected to the MYSEA Server via TCMs and link encryptors. TCMs complement link encryption by ensuring proper labeling of data passed back to the MYSEA Server.
- Single level servers in the Multilevel Enclave area provide application services to both local clients and those in the legacy networks. Intrusion detection systems which monitor activity on single-level networks are examples of these.

2.2 Dynamic Security Services (DSS)

Complex and adaptive networks may require demand-driven changes to the security provided. When conditions on the network change, requirements for security—e.g., restrictions as seen from the users' or attackers' point of view—may also change (Levin 2006). In MYSEA, the DSS Quality of Security Service (QoSS) mechanisms located on the TPE and at the MYSEA Server can modulate the protection services afforded to an ongoing user session, in response to a change notification. Protection mechanisms for client-server communications may be based upon network conditions such as INFOCON mode. This allows the quality of protection for the entire MLS LAN to be modified from a single point of administrative control within the high-assurance federation, dynamically, based on environmental conditions.

2.3 MLS policy-constrained application services

Policy-constrained applications may be invoked on the MYSEA Server from client workstations, as well as from other components on the server. *Policy-constrained* means that an application has been

modified to run in a multilevel environment without requiring extraordinary privileges, so it is both fully functional and constrained by the security policy (Irvine 1990). MYSEA currently supports a number of application protocols (HTTP, IMAP, etc) and provides many policy-constrained application services including web, email, wiki, webmail, WebDAV, HTTP streaming video, and VoIP services.

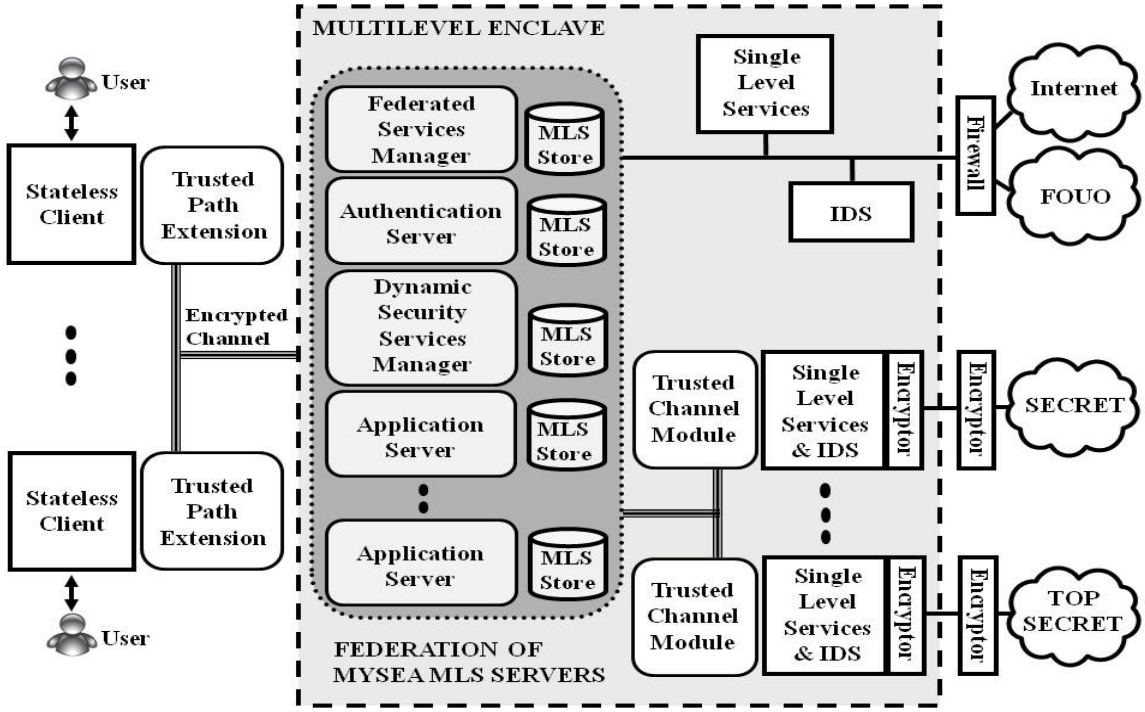

Figure 1: Monterey Security Architecture

2.4 Client usage scenario

When a user logs in, the TPE passes the user's identity credentials to the MYSEA Server, which validates the login attempt and instructs the TPE whether to allow or deny access to the network. In negotiating a session level, the TPE passes the user's session level request to the Server for a decision. After a successful login and session level negotiation, the TPE allows the user to access the MLS LAN, the MYSEA Server and its services, as well as the single-level networks.

At any time, the user can invoke the trusted path to request a session level change, log off, etc. The Trusted Path Extension blocks access to the network while the user's security attributes are in flux during such operations. To meet object reuse requirements (NCSC 1992), client state is purged at the end of each session, and data created or modified on the clients is stored on the MYSEA Server.

3. Concept of operations

We describe the concept of operations for a cross-domain intrusion detection system (CD-IDS) in the context of MYSEA.

Using the TPE, a network security analyst logs in to MYSEA and begins a user session at a classification level within her clearance (*e.g.*, a session at SECRET). The analyst uses her thin client's web browser to access the CD-IDS analysis application running on the MYSEA server. Via this web-based interface, the analyst is able to review reports and data collected by the CD-IDS monitoring components deployed on any network associated with a level that is dominated by her session level (*e.g.*, those alerts or data reported by the monitors on the UNCLASS and SECRET networks).

In particular, the analyst is *not* able to access data or reports associated with a network above her session level (*e.g.*, she cannot read alerts or data associated with a TOP SECRET network) nor is she able to access those networks not equal to her clearance (*i.e.*, the analyst cannot access resources on the UNCLASS network during her SECRET session).

The analyst may choose to save any report generated by the CD-IDS analysis application or to create new analysis artifacts based on the information obtained from the IDS reports. The analysts may share reports with the system's security administrator or with other analysts, by leveraging existing MYSEA features. For example, she may save reports to her "home" directory on the MYSEA server via WebDAV access, or email the report to another user. In each situation, the report is written at the level of her current session (*e.g.*, SECRET) and normal MAC information-flow policies are enforced. To terminate the session, the analyst logs out and her thin client is purged of all state.

Responding to the alerts and analysis reports is handled through a different process and is outside the scope of CD-IDS. It is noted, however, that there are number of possible actions that may be taken by the system's security administrator in response to a perceived attack, *e.g.* she may change the current network protection policy to a more restrictive one using DSS. This potential enhancement is further elaborated in Section 5.

4. CD-IDS implementation

Before discussing the design and implementation of the CD-IDS prototype, we describe the high-level requirements for the software components of the system.

4.1 Software requirements

The following requirements are derived from popular secure design principles, or inherited from the MYSEA project:

- *Minimize the introduction of new security-relevant or privileged components*—in particular, software should defer to the authentication and single sign-on mechanisms provided by MYSEA. This requirement is motivated by the desire to minimize the size and complexity of the MYSEA Trusted Computing Base for evaluation purposes (CCMB 2006);
- *Software should not require any client-state be maintained*—MYSEA clients purge state on user session termination;
- *Third-party source code must be available and its license must allow modification*—MYSEA uses system call interposition with trusted proxies to enable unprivileged software to access the MLS LAN. This, currently, involves slight modification of the software to run on the MYSEA platform;
- *Software cost should be zero or very low*—this follows the MYSEA goal to provide a system whose cost of ownership is low;
- *Interfaces should be intuitive and user friendly*—this follows the MYSEA objective of providing a familiar, productive work environment, in which system behaviors are consistent with user expectations and in which the principles of ergonomic security are obeyed.

4.2 Functional requirements

The CD-IDS must meet the following functional requirements:

- *The CD-IDS shall make use of existing monitoring components (i.e., single-level IDS monitors);*
- *The CD-IDS shall provide a mechanism for each single-level IDS monitor to store alerts and other data on the MYSEA Sever in a database that is classified at the level of the network to which the monitor is attached;*
- *The CD-IDS shall allow users operating at different session levels to view alerts stored at or below their session level;*
- *The CD-IDS shall deny any user request to modify the data collected by the single-level IDS monitors;*
- *The CD-IDS shall provide a web-based user interface that can present a consolidated view of collected data and write data only at the user's session level;*
- *The CD-IDS shall keep a log of all user requests.*

4.3 Design considerations

Following these requirements, an information flow design for the CD-IDS prototype in the context of MYSEA was developed.

In particular, Figure 2 illustrates the information flow between an IDS monitor on a single-level network, the single-level database in which the monitor stores its alerts, the IDS analysis engine that presents the MLS view of the collected data via a web-based interface, and the analyst's workstation. Each single-level database runs at the level of the network for which it provides service. For an analyst running at SECRET who requires read-access to the database service at UNCLASS, some component requires both (1) read-access to a subject at SECRET (the software running on behalf of the analyst) and (2) the ability to query (*i.e.*, read-write access to) the database service available on the UNCLASS network; thus, it necessarily follows that this component must be trusted, and mediate this access appropriately.

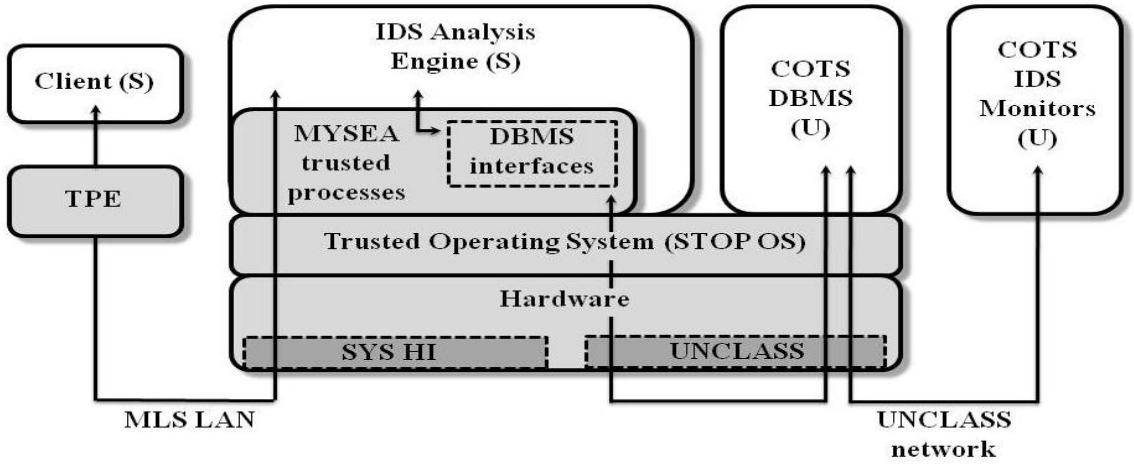

Figure 2: CD-IDS information flow

Abstractly, Figure 2 is reminiscent of the Woods Hole architecture (NAS 1986) in which a trusted interface emulates a full MLS database using multiple, single-level databases. Indeed, this type of trusted, distributed database architecture could be used in this situation. Traditional Woods Hole, however, requires the trusted interface mediate all access to the data stores (to mediate write-conflicts, handle poly-instantiation, etc); thus, for CD-IDS, we would require a variant of Woods Hole that provides trusted services on all network interfaces, i.e., mediating all requests to the set of single-level databases including those from the analyst on the MLS LAN and those from the monitors on each single-level network. We, however, view a single, monolithic, trusted component with simultaneous read-write access to all networks to be highly undesirable. Thus, as an alternative, we have opted for a design using small, trusted database proxies, each of which mediates access to a single database. We elaborate on how this design decision was implemented for the CD-IDS prototype below.

4.4 Prototype implementation

The CD-IDS prototype consists of a policy-aware IDS analysis engine application running on MYSEA. Building on MYSEA's existing support for Apache and PHP, this application presents an MLS view of the alerts, logs and other data collected by the single-level IDS monitoring components on each classified network attached to MYSEA. The main components of the CD-IDS prototype are summarized below.

4.4.1 Single-level IDS monitors

These components reside on each single-level network attached to MYSEA. Each component collects traffic data and generates alerts for possible attacks on its respective network. The monitors send their alerts and logs to companion databases that reside on the MYSEA Server. The IDS monitors are not trusted to enforce MLS policy.

In the CD-IDS prototype, the open source intrusion detection software Snort (Snort n.d.) implements this component.

4.4.2 Single-level DBMS

A single-level database management system (DBMS) manages the databases used by the single-level IDS monitoring components for a network. There are multiple instances of this component, one per security level. The DBMS is not trusted and runs at the security level of the network for which it is configured to provide database services.

In the CD-IDS prototype, this component is implemented by the open source object-relational database system PostgreSQL 0).

4.4.3 Intrusion analysis engine

The intrusion analysis engine provides the user interface to access and analyze the alerts associated with each network. This component is not trusted and runs at the user's session level. This application presents the analyst with a web-based interface to accessing this data.

In the CD-IDS prototype, the open source Basic Analysis and Security Engine (BASE) (BASE n.d.) implements this component. BASE was selected because it is a popular analysis tool for Snort. BASE was slightly modified to support cross-domain analysis functionality, including the addition of logic to display an advisory classification level for the analyst and logic to query multiple databases. For the prototype, BASE is utilized by leveraging MYSEA's existing support for PHP and Apache as policy-constrained untrusted applications.

4.4.4 Database Proxy Service (DBPS)

The Database Proxy Service mediates access to the single-level DBMS. There are multiple DBPS instances, one per security level. The DBPS is trusted and has special privileges to read from and write to resources of different security levels. The DBPS primarily acts as a guard, by selectively dropping or proxying SQL commands to the DBMS. In particular, the DBPS enforces the following policy:

- Proxy all requests, *e.g.* SELECT, UPDATE, DELETE, INSERT, if the security level of the requestor (*i.e.*, the level of analyst's user session) is equal to the security level of the target (*i.e.*, the security level of the destination DBMS);
- Allow requests for the SELECT command if the security level of the requestor dominates the security level of the target;
- Disallow all requests if the security level of the requestor does not dominate the security level of the target.

In the CD-IDS prototype, the open source database middleware program pgpool-II (pgpool-II n.d.) was modified to implement this component.

4.4.5 Secure Session Service (SSS)

The Secure Session Service (SSS) is a MYSEA trusted process that mediates access to the MLS LAN. Thus, the SSS handles communication between BASE and the analyst's workstation, and between BASE and the DBPS (which is, itself, deployed as a service on the MLS LAN interface). The SSS is trusted and has the same essential privileges as the DBPS, though its responsibilities are different.

The SSS exports a virtualized socket interface—resembling typical Unix system calls—to the untrusted subjects on the MYSEA Server. By using these interfaces, network requests are proxied to the SSS, which then performs the requests on the untrusted subject's behalf. The SSS enforces policy by inspecting each of these network requests and determining whether to perform them or report failure; decisions are based on the client's active session level, the levels of the source and destination for the network request, etc.

Next, we describe how each of these components communicate and function together to implement the CD-IDS prototype.

Figure 3 illustrates the processing flow and interactions among the CD-IDS components when an analyst, after starting a user session at the SECRET level, asks for IDS alerts stored in the SECRET and UNCLASS databases.

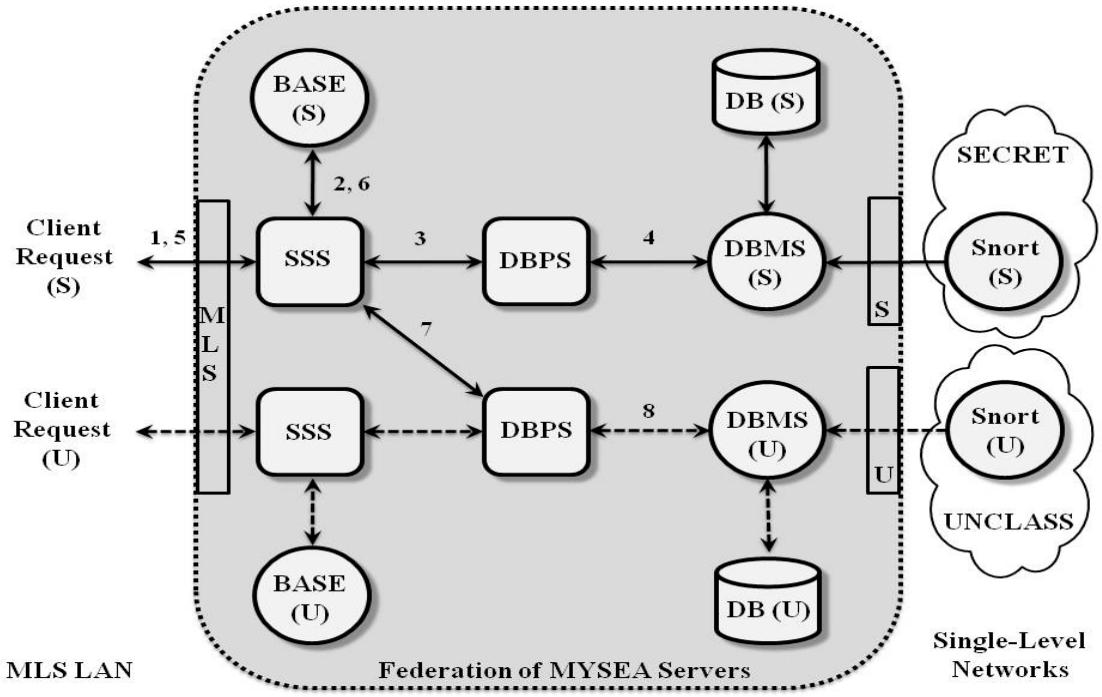

Figure 3: CD-IDS prototype design

1. The client's browser requests IDS alert data for the SECRET network via the web interface provided by BASE.

2. BASE requests the SSS make a connection to the DBPS, since BASE lacks the privileges requires to establish a direct socket connection to communicate with either clients on the MLS LAN or the DBMS.

3. The SSS establishes a connection to the DBPS associated with the DBMS at SECRET (but not the DBMS itself). The DBPS will forward all subsequent requests for SECRET network alerts to the DBMS at SECRET. Note that neither BASE nor the DBMS is aware that it is communicating with a proxy; for BASE, the DBPS acts as a server while, for the DBMS, the DBPS acts as a client.

4. The DBPS establishes a connection to the DBMS at SECRET. For each query, the DBPS enforces the policy rules described earlier. In particular, as the security level of both the target and requestor is SECRET, the DBPS will forward all database queries. The result of each query is passed back to the client via the DBPS and SSS.

5. The client's browser requests IDS alert data for the UNCLASS network.

6. As in Step 3, the SSS establishes a connection to the DBPS associated with the DBMS at UNCLASS.

7. The SSS forwards all requests for alert data on the UNCLASS network to the DBMS at UNCLASS via the UNCLASS DBPS. The DBPS establishes a connection to the DBMS at UNCLASS.

8. Since the requestor's session level is SECRET and the security level of the target DBMS is UNCLASS, the DBPS forwards only specific, read-only requests, *i.e.* SELECT, to the DBMS at a lower classification level. The results of the query are passed back to the browser through via the DBPS and SSS.

Figure 3 also shows a similar processing flow, indicated by dashed arrows. In this parallel case, another analyst at an UNCLASS session level uses the CD-IDS to view alerts associated with the

UNCLASS network, following the process outlined in Steps 1–4. Although not illustrated, attempts to access the SECRET DBMS are denied by the DBPS associated with the UNCLASS DBMS.

5. Future work

In this section, we describe some possible improvements to the CD-IDS prototype of particular interest to the project, including extending support for other COTS IDS products and developing a unified framework for network threat-response.

5.1 Covert channel analysis

The system's intrusion analysis engine application (BASE) always requests a connection to the single-level DBMS before issuing it a query. The DBPS forwards the connection request to the IDS database, as the current prototype DBPS only filters SQL commands. Thus, the presence or lack of presence of a connection perceived by a low subject (the DBMS) may be modulated by a subject at a higher level (the analysis engine application). This modulation may potentially be exploited as a covert channel. This channel can be closed if each proxy were to establish and maintain a persistent connection to its IDS database. This would effectively limit the possible communication channels from high to low that make use of the DBPS guard to those channels utilizing the SELECT requests—which are, of course, necessary for proper operation of the system.

5.2 Integrating automated threat response

One possible enhancement to the CD-IDS prototype is to provide automated responses to perceived attacks, *i.e.* turning the CD-IDS prototype into a cross-domain intrusion prevention system (CD-IPS); initially for the MYSEA LAN and, in the future, for the single-level networks. The latter may employ common IPS technologies and techniques (DoC 2003) to reduce the impacts of the attacks to an acceptable level.

An example for the MYSEA LAN scenario is as follows: In the presence of an attack on the UNCLASS network, the CD-IPS may change the active dynamic security services protection policy to one which prevents non-administrative MYSEA users from using network services during UNCLASS sessions (thus, preventing users from falling victim to attacks on that network). Such work would include developing a policy language appropriate for representing DSS network protection policies, developing flexible and customizable rules for when to invoke certain types of policies, and developing a scriptable interface or web interface for DSS administrative functions.

Figure 4 depicts one possible design describing this capability.

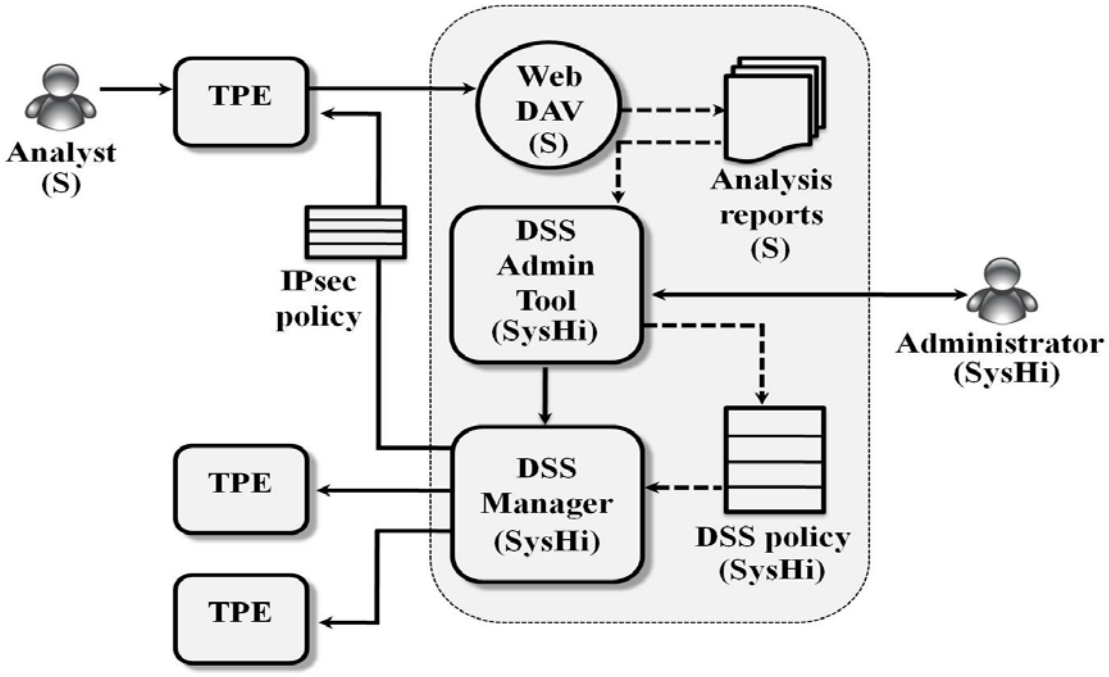

Figure 4: Notional CD-IPS design

After assessing the collected IDS data from BASE (not shown), the analyst creates and files intrusion analysis reports via WebDAV; the reports are kept in an "intrusion analysis data store" and tagged based on the criticality of the perceived threat. The DSS Administrative Tool (DSS-AT) monitors the data store and notifies the system's security administrator when one or more high-criticality reports are submitted. Using the DSS-AT, the system administrator reviews the analysis reports and, if the threat conditions warrant a change of the active DSS policy, generates a more restrictive DSS policy. The administrator then instructs the DSS Manager, via the DSS-AT, to switch to the new DSS policy. The DSS Manager pushes the new DSS policy in the form of IPsec policy rules to individual TPEs; the IPsec policy rules are automatically generated by the DSS-AT. This approach provides the key functionalities associated with IPS technologies (DoC 2003).

5.3 Support for other IDS monitors

Suricata (Suricata n.d.) is an open source intrusion detection and prevention engine recently released by the Open Information Security Foundation. Some features that differentiate Suricata from Snort include full IPv6 support and multi-threading. Since Suricata writes its data to a spool in the unified/unified2 format, it should be possible to write its spool data (*i.e.*, its alerts) to the same DBMS backend utilized by BASE, using barnyard/barnyard2 (Barnyard n.d.). Thus, integrating Suricata—or, in general, any legacy IDS using the unified/unified2 format—into the CD-IDS is possible. Exploring the extent of this support, however, requires further exploration.

6. Conclusion

In this paper, we introduced the problem of monitoring and analyzing detected intrusion attempts across multiple single-level networks. We described our view of a cross-domain network-based intrusion detection system, in which a security analyst can obtain a coherent view of the intrusion alerts collected by the separate IDS monitors on all those single-level networks for which she has authorized access. We described an application-level approach for providing a CD-IDS that utilizes the COTS IDS monitors already present in single-level network enclaves, while leveraging various open-source products. We described the high-level requirements and design of our CD-IDS prototype, as well as its high-assurance MLS computing platform, *viz.*, MYSEA. The implementation of the CD-IDS prototype was described, with special attention to the interactions among the trusted and untrusted components of the system. The paper concluded with several improvement features, including the ability to dynamically alter network protection settings based on the perceived threat conditions.

Acknowledgements

This work was sponsored in part by the Office of Naval Research and the National Reconnaissance Office. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsoring organizations.

References

BAE Systems Information Technology Inc. (2004) "Security Target, Version 1.11 for XTS-400 Version 6", December.

Bailey, M. (2007) "The unified cross domain management office: bridging security domains and cultures," *CrossTalk magazine*, vol. 21, no. 7, pp. 21–23, July.

Barnyard project (n.d.). Available at: http://sourceforge.net/projects/barnyard [Accessed March 2011].

Basic Analysis and Security Engine project (n.d.). Available at: http://base.secureideas.net [Accessed March 2011].

Common Criteria Maintenance Board (2006) "Common Criteria for Information Technology Security Evaluation," CCMB-2006-09-001, Version 3.2 revision 1 edition, September.

Denning, D. E., Lunt, T. F., Schell, R. R., Shockley, W. and Heckman, M. (1988) "The SeaView security model," in *Proc. of IEEE Symposium on Security and Privacy*, pp. 218–233, April.

Irvine, C. E., Acheson, T. and Thompson, M. F. (1990) "Building Trust into a Multilevel File System," in *Proc. 13th National Computer Security Conference*, October, pp. 450–459.

Irvine, C.E., Nguyen, T. D., Shifflett, D. J., Levin, T. E., Khosalim, J., Prince, C., Clark, P.C. and Gondree, M. (2009) "MYSEA: the Monterey security architecture," in *Proc. of the Workshop on Scalable Trusted Computing (ACM STC), Conference on Computer and Communications Security (CCS),* Association for Computing Machinery (ACM), Chicago, Illinois, November, pp. 39–48.

Levin, T. E., Irvine, C. E. and Spyropoulou, E (2006) "Quality of Security Service: Adaptive Security," Volume 3, John Wiley and Sons, Hoboken, NJ, January, pp.1016-1025.

National Computer Security Center (1992) "A guide to understanding object reuse in trusted systems," Technical Report NCSC TG-018, Fort George G. Meade, MD, July.

pgpool-II project (n.d.). Available at: http://pgpool.projects.postgresql.org. [Accessed March 2011].
PostgreSQL project (n.d.). Available at: http://www.postgresql.org. [Accessed March 2011].
Snort project (n.d.). Available at: http://www.snort.org. [Accessed March 2011].
Suricata project (n.d.). Available at: https://redmine.openinfosecfoundation.org/projects/suricata. [Accessed March 2011].
National Academy of Sciences (1986) *"The 1986 Workshop on Integrated Database Development for the Building Industry*," Woods Hole, MA, National Academy Press, Washington, D.C., June.
U.S. Department of Commerce (2007) "Guide to Intrusion Detection and Prevention Systems (IDPS)," NIST Special Publication 800-94. National Institute of Standards and Technology, Gaithersburg, MD, February.
U.S. Department of Defense (2003) "Instruction Number 8500.2, Information Assurance (IA) Implementation," February.
BAE – see BAE Systems Information Technology Inc.
BASE – see Basic Analysis and Security Engine
CCMB – see Common Criteria Maintenance Board
DoC – see U.S. Department of Commerce
DoD – see U.S. Department of Defense
NAS – see National Academy of Sciences
NCSC – see National Computer Security Center